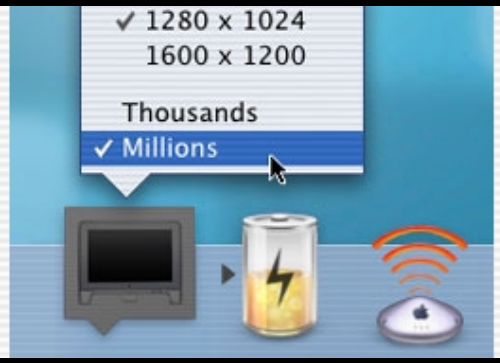
Session 303

Network Configuration and Mobility



Thomas Weyer
Networking & Communications
Technology Manager

Session 303

Network Configuration and Mobility

Dieter Siegmund
Software Engineer

Introduction

- Network configuration and mobility architecture
- Work in progress
- Your feedback

Network Configuration/Mobility

- Goals:
  - Support dynamic network configuration changes
  - Allow multiple network interfaces simultaneously
  - Minimal user intervention when possible

Network Configuration/Mobility

- Example: laptop used at home and work
  - Use network at home, then put system to sleep
  - At work, connect cable and wake system up
  - Network is reconfigured automatically, or user changes “locations”
- Example: laptop uses both LAN and AirPort
  - System prefers LAN over AirPort while LAN is connected
  - When LAN disconnects, system switches to AirPort

Network Configuration/Mobility

- Requirements
  - Low-level network events (e.g., link status changes)
  - Persistent storage for network configurations
  - Dynamic storage for network state information
  - Mechanism to apply configuration changes
  - Flexible schema in permanent and dynamic storage to allow for future expansion

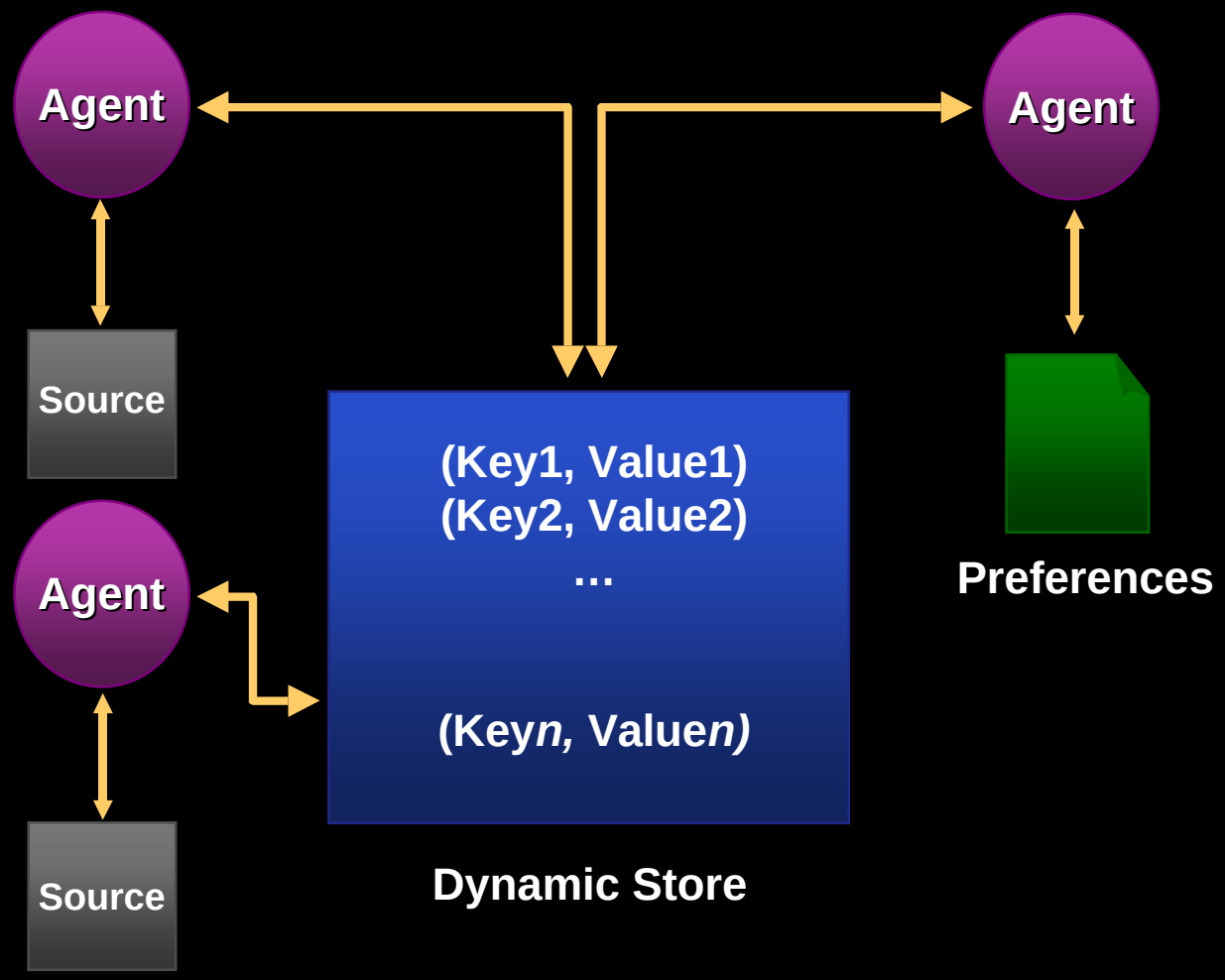
Network Configuration/Mobility

- System Configuration Architecture
  - Generalize beyond Network Configuration
- Leverage existing technology
  - CoreFoundation framework
- Darwin
  - Open Source

System Configuration Architecture

- Dynamic Store
  - Stores current system configuration information
  - Provides notification services
- System Configuration Preferences
  - Persistent storage for configuration information
- Configuration Agents
  - Apply configuration to the system
  - Interact with the Dynamic Store

System Configuration Architecture



Dynamic Store

- Contains current system configuration information
- Not persistent across reboots
- A dictionary, i.e., a set of (Key, Value) pairs
- Key is a CFString whose format is a hierarchical namespace (like a URL):

  <domain> <path>

  e.g. State: /Network/Global/DNS
- Value is a CFPropertyList, usually a CFDictionary

Dynamic Store

- A client registers interest in particular keys and receives a notification when any of those keys change
- Notifications are “Level-triggered” not “edge-triggered”
- Provided by System Configuration Server, configd
- Heart of the mobility architecture

System Configuration Preferences

- Persistent storage for system setup information
- Like CFPreferences but with notifications
- Accessible from system daemons
- Darwin
- Uses CFPropertyList types, serialized as XML
- Stored in:
  /var/db/SystemConfiguration/preferences.xml

System Configuration Preferences

- Defined preferences
  - CurrentSet
  - NetworkServices
  - Sets
  - System

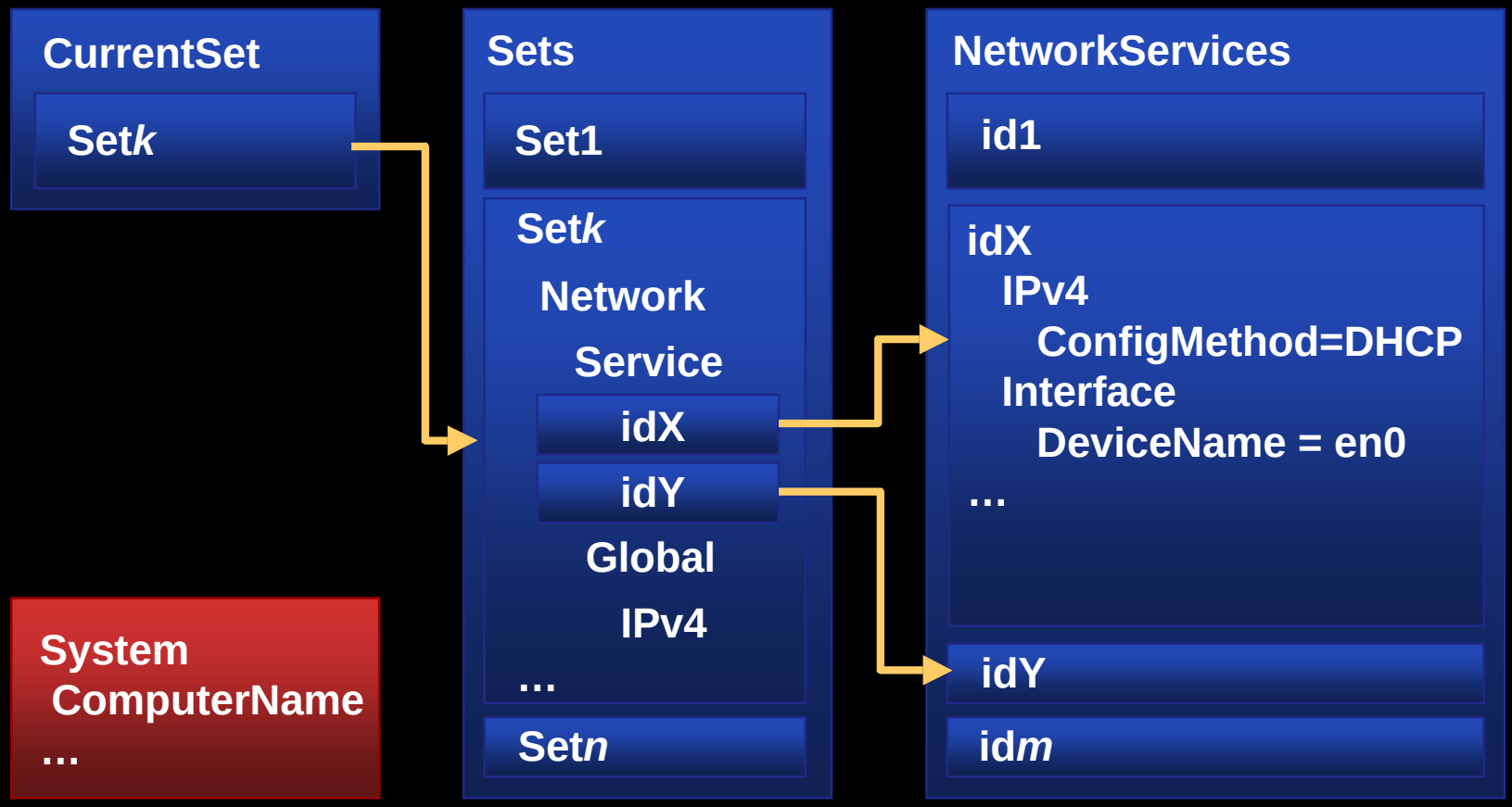
System Configuration Preferences

- CurrentSet
  - A reference/link to the currently active set
- NetworkServices
  - The complete list of Network Services defined across all sets
- Sets
  - The list of sets
- System
  - Configuration that is not location-specific, e.g., the Computer Name

System Configuration Preferences

- Set
  - The complete configuration for a single “location”
  - Contains a list of Network Service references/links, the order of the Network Services, and other network global configuration
- Network Service
  - A set of Network Entities for a single network interface or network connection
  - Identified by a unique Network Service ID
- Network Entity
  - A list of properties for a single protocol or area of interest

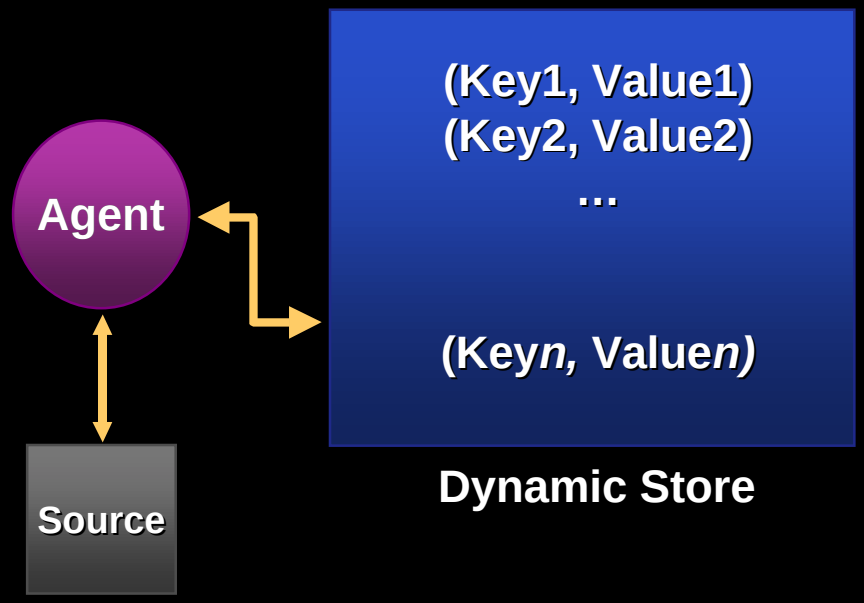
System Configuration Preferences



Configuration Agent

- Manages some aspect of the configuration
- Associated with one or more configuration sources
- Well-defined role

Configuration Agent

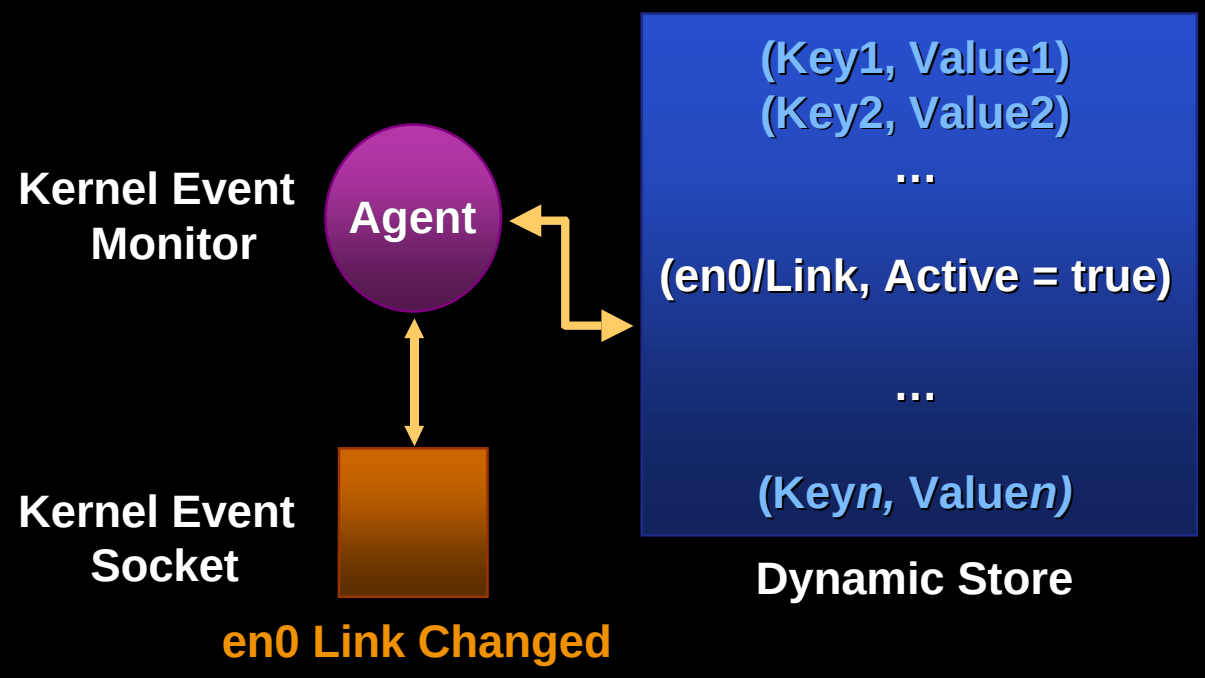


Configuration Agent

- Agents
  - Kernel Event Monitor
  - Preferences Monitor
  - IP Configuration (DHCP/BOOTP client)
  - IP Monitor (manages the default route, DNS)
  - PPP Controller (dial-up, PPPoE)

Configuration Agent

Kernel Event Monitor



Kernel Event Monitor

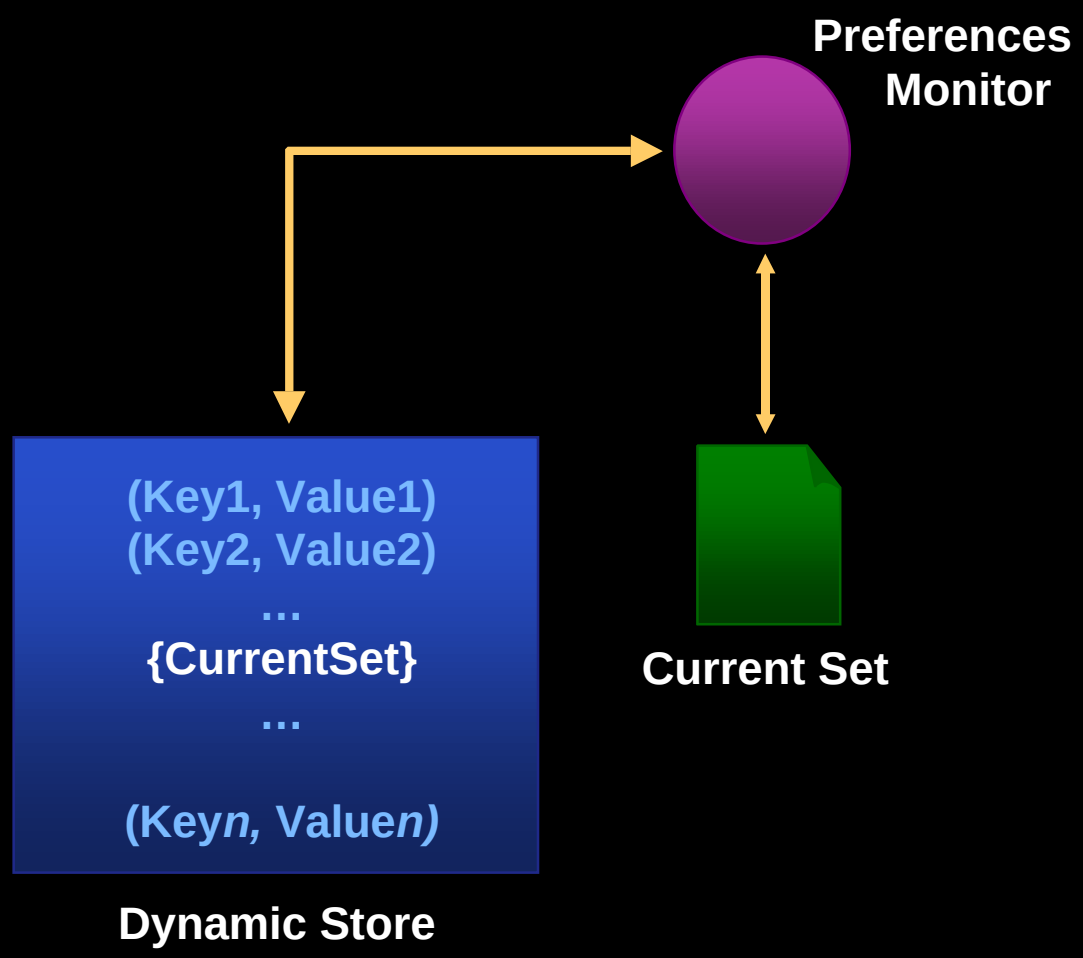
- Maintains network interface state
- Source is a kernel event socket
- Maintains a list of network interface names
  - Key **State:/Network/Interface**
  - Dictionary contains an “Interfaces” property, an array of strings
- The link status of each interface
  - Key **State:/Network/Interface/<ifname>/Link**
  - Dictionary contains an “Active” property, a boolean

Kernel Event Monitor

- Generates notifications for other agents
  - IP Configuration agent

Configuration Agent

Preferences Monitor



Preferences Monitor

- Configuration source is System Configuration Preferences
- Maps CurrentSet dictionary into the Dynamic Store
- Flattens hierarchical dictionary into single level

Preferences Monitor

Dictionary Before Flattening

```
"Service X" = (dict) {
    "UserVisibleName" = (string) "Work LAN";
    "DNS" = (dict) {
        "ServerAddresses" = (array) {
            (string) "17.128.100.12",
            (string) "17.128.100.10"
        }
        "DomainName" = (string) "apple.com";
    }
}
```

Preferences Monitor

Dictionary Before Flattening

```
"/Service X" = (dict) {
    "UserVisibleName" = (string) "Work LAN";
}
"/Service X/DNS" = (dict) {
    "ServerAddresses" = (array) {
        (string) "17.128.100.12",
        (string) "17.128.100.10"
    }
    "DomainName" = (string) "apple.com";
}
```

Preferences Monitor

- Traverses dictionary "tree" starting at root
  - At each dictionary "node", computes a **Key** and **Value**
  - **Key** is the "path" of the dictionary
  - **Value** is dictionary "node" with sub-dictionaries removed
  - Resolves references/links
  - Eliminates inactive and empty dictionaries
- Attaches **Setup:** domain prefix to each **Key**

Preferences Monitor

- Provides finer-grained (per dictionary) notifications
  - When preferences changes, generates new map
  - Computes difference between new map and old, applies difference to Dynamic Store
- Single agent reads/understands the preferences layout
- Other agents need only know the Dynamic Store layout

IP Configuration Agent

- Configures Ethernet interfaces for IP (Manual, DHCP, BOOTP)
- Reads configuration from Setup:
- Notified when keys change
- Writes Dynamic Store State: domain keys when configuration is applied
  - IP address and subnet mask
  - Adds Router and DNS information from BOOTP/DHCP

IP Configuration Agent (Cont.)

- If link remains inactive for some period (4 secs), de-configures
- When link becomes active, retries configuration method

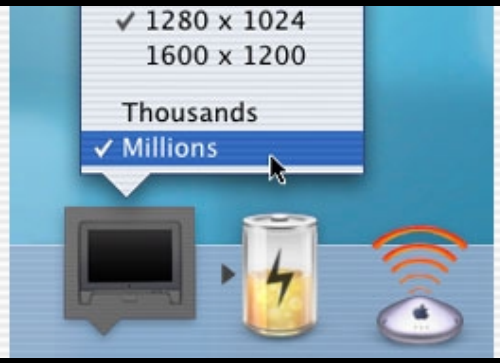
PPP Controller Agent

- Configures PPP interfaces for IP (dial-up, PPPoE)
- Reads configuration from Setup:, notified when keys change
- Writes Dynamic Store State: domain keys when configuration is applied
  - IP address, destination IP address, router
  - Adds DNS information returned from PPP server
- Creates PPP interfaces as needed
- Interacts with dialers

IP Monitor Agent

- Selects a service/interface as the “primary”
- The “primary” service/interface is the one associated with the default route and DNS for the system
- Watches changes made by service/interface configuration agents
  - IP Configuration and PPP Controller
- “ServiceOrder” array decides which service to select when multi-homed
- Driven completely by notifications from **State:** domain

Demo



Allan Nathanson
Software Engineer

System Configuration Framework

- Contains mostly low-level APIs for:
  - Dynamic Store
  - Preferences
- Some higher-level APIs
  - Reachability APIs to test whether system will dial
- Currently private
  - Working on making it public

System Configuration Framework

- Need more higher level APIs
  - Avoid embedding schema-specific knowledge in developers' code
  - Avoid duplicating code
- Evolve and grow as needs arise

Related Sessions

300 Networking Overview: Mac OS X network protocols and APIs	Room A1 **Tues., 9:00am**
138 Core Foundation Overview: Information on CF APIs	Civic **Fri., 9:00am**

Who to Contact

Thomas Weyer
Network & Communications Technology Manager
Apple Worldwide Developer Relations
weyer@apple.com

For More Information

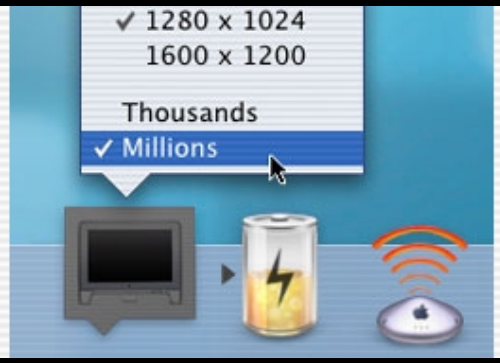
Darwin

http://www.opensource.apple.com

Projects

configd	Dynamic Store, Preferences APIs System Configuration Server (configd)
configd_plugins	IP Monitor, Preferences Monitor
bootp	IP Configuration
ppp	PPP Controller

Q&A



Thomas Weyer
Network & Communications Technology Manager
weyer@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001